Bài tập thực hành:

BTTH-01: CÁC TOÁN TỬ HÌNH THÁI HỌC

----------o0o----------

Họ và tên: Nguyễn Phan Quốc Bảo

Mã số sinh viên: 19120456

Email: 19120456@student.hcmus.edu.vn

SĐT: 0368916545

1. Nội dung bài tập:

STT	Yêu cầu	Mức độ hoàn thành
1.	Toán tử giãn nở nhị phân (Binary Dilation)	100%
2.	Toán tử co nhị phân (Binary Erosion)	100%
3.	Toán tử mở nhị phân (Binary Opening)	100%
4.	Toán tử đóng nhị phân (Binary Closing)	100%
5.	Toán tử Hit-or-Miss	100%
6.	Thinning	100%
7.	Toán tử giãn nở độ xám (Grayscale Dilation)	100%
8.	Toán tử co độ xám (Grayscale Erosion)	100%
9.	Toán tử mở độ xám (Grayscale Opening)	100%
10.	Toán tử đóng độ xám (Grayscale Closing)	100%
11.	Toán tử Morphological Reconstruction độ xám	0%
12.	Toán tử Morphological Gradient độ xám	100%
13.	Toán tử Top-Hat	100%
14.	Toán tử Black-Hat	100%
15.	(mở rộng) toán tử Boundary Extraction nhị phân	100%
16.	(mở rộng) Toán tử textural Segmentation độ xám	100%

2. Cài đặt các toán tử

2.1. Toán tử giãn nở nhị phân (Binary Dilation)

Đưa mặt nạ kernel dò trên toàn bộ ảnh, khi gặp những điểm ảnh có giá trị là 1 thì in vết của kernel lên trên ảnh

$$X \oplus B = \{p \in \varepsilon^2 | p = x + b, x \in X \text{ và } b \in B\}$$

2.2. Toán tử co nhị phân (Binary Erosion)

Đưa mặt nạ kernel dò trên toàn bộ ảnh, khi gặp vị trí giá trị vùng điểm ảnh và kernel hoàn toàn giống nhau thì trả về 1, ngược lại trả về 0.

$$A \Theta B = \{p \in \varepsilon^2 | x + b \in X, \forall b \in B\}$$

2.3. Toán tử mở nhị phân (Binary Opening)

Thực hiện erosion ảnh gốc với kernel rồi tiếp tục dilation ảnh kết quả với kernel

$$\mathrm{A} \circ \mathrm{B} = (A \Theta B) \oplus B$$

2.4. Toán tử đóng nhị phân (Binary Closing)

Thực hiện dilation ảnh gốc với kernel rồi tiếp tục erosion ảnh kết quả với kernel

$$\mathrm{A} \bullet \mathrm{B} = (A \oplus B) \Theta B$$

2.5. Toán tử Hit-or-Miss

Từ 1 kernel lưu cả thông tin của kernel hit và kernel miss, tách thành 2 kernel: kernel hit (B1), kernel miss(B2). Thực hiện erode ảnh A với kernel B1, thực hiện erode ảnh nghịch đảo của A với kernel B2 và thực hiện lấy phần giao 2 kết quả, ta thu dược ảnh Hit or miss:

0	1	0
1	-1	1
0	1	0

Kernel lưu thông tin chung

0	1	0
1	0	1
0	1	0

B1

0	0	0
0	1	0
0	0	0

B2

$$X \circledast B = (X \Theta B_1) \cap (X^C \Theta B_2)$$

2.6. Thinning

Thực hiện hit or miss với 8 mặt nạ cấu trúc sau:

-1	-1	-1
0	1	0
1	1	1

B1

0	-1	-1
1	1	-1
1	1	0

B2

1	0	-1
1	1	-1
1	0	-1

B3

1	1	0
1	1	-1
0	-1	-1

B4

1	1	1
0	1	0
-1	-1	-1

B5

0	1	1
-1	1	1
-1	-1	0

B6

-1	0	1
-1	1	1
-1	0	1

B7

-1	-1	0
-1	1	1
0	1	1

B8

Lặp lại quá trình trên cho đến khi ảnh của vòng lặp sau không thay đổi so với vòng lặp trước.

$$A \otimes B = A \cap (A \circledast B)^C$$

$$\{B\} = \{B_1, B_2, \dots, B_8\}$$

$$A_k \otimes \{B\} = ((\dots((A_{k-1} \otimes B_1) \otimes B_2) \dots) \otimes B_8)$$

2.7. (mở rộng) toán tử Boundary Extraction nhị phân

Toán tử trích biên, rút trích biên cạnh của đối tượng bằng cách lấy ảnh gốc trừ cho ảnh erosion.

$$\beta(A) = A - (A \Theta B)$$

2.8. Toán tử giãn nở độ xám (Grayscale Dilation)

Xét vùng lân cận 1 pixel và gán giá trị mới cho pixel đó bằng giá trị lớn nhất của vùng lân cận cộng cho pixel trên kernel tương ứng.

$$(f \oplus b)(s,t) = \max\{f(s-x, t-y) + b(x,y) | (s-x), (t-y) \in D_f; (x,y) \in D_b\}$$

2.9. Toán tử co độ xám (Grayscale Erosion)

Xét vùng lân cận 1 pixel và gán giá trị mới cho pixel đó bằng giá trị bé nhất của vùng lân cận trừ cho pixel trên kernel tương ứng.

$$(f \Theta b)(s,t) = \min\{f(s-x, t-y) - b(x,y) | (s-x), (t-y) \in D_f; (x,y) \in D_b\}$$

2.10. Toán tử mở độ xám (Grayscale Opening)

Thực hiện erosion ảnh gốc với kernel rồi tiếp tục dilation ảnh kết quả với kernel

$$f \circ b = (f \Theta b) \oplus b$$

2.11. Toán tử đóng độ xám (Grayscale Closing)

Thực hiện dilation ảnh gốc với kernel rồi tiếp tục erosion ảnh kết quả với kernel

$$f \bullet b = (f \oplus b)\Theta b$$

2.12. Toán tử Morphological Gradient độ xám

Thực hiện dilation ảnh gốc và erosion ảnh gốc, sau đó lấy 2 ảnh này trừ nhau, ta được ảnh gradient độ xám.

$$h = (f \oplus b) - (f\Theta b)$$

2.13. Toán tử Top-Hat

Thực hiện Opening ảnh gốc, sau đó lấy ảnh gốc trừ ảnh đã opening

$$h = f - (f \circ b)$$

2.14. Toán tử Black-Hat

Thực hiện Closing ảnh gốc, sau đó lấy ảnh gốc trừ ảnh đã closing

$$h = f - (f \bullet b)$$

2.15. (mở rộng) Toán tử textural Segmentation độ xám

Thực hiện closing với kernel b1, sau đó thực hiện opening với kernel b2

$$h = (f \bullet b_1) \circ b_2$$

3. Cách chạy source code

Các tham số dòng lệnh đều có dạng:

python main.py -i <đường dẫn ảnh input> -o <đường dẫn ảnh output> -p <mã toán tử> -t <thời gian chờ>

Ví dụ:

```
python main.py -i Lenna.png -o Lenna2.png -p dilate -t 100
```

Các thông số của kernel được sửa đổi thông qua source code

Các mã toán tử:

Toán tử giãn nở nhị phân (Binary Dilation): dilate

Toán tử co nhị phân (Binary Erosion): erode

Toán tử mở nhị phân (Binary Opening): open

Toán tử đóng nhị phân (Binary Closing): close

Toán tử Hit-or-Miss: hitOrMiss

Thinning: thinning

(mở rộng) toán tử Boundary Extraction nhị phân: boex

Toán tử giãn nở độ xám (Grayscale Dilation): GSdilate

Toán tử co độ xám (Grayscale Erosion): GSerode

Toán tử mở độ xám (Grayscale Opening): GSopen

Toán tử đóng độ xám (Grayscale Closing): GSclose

Toán tử Morphological Gradient độ xám: GSgradient

Toán tử Top-Hat: GStophat

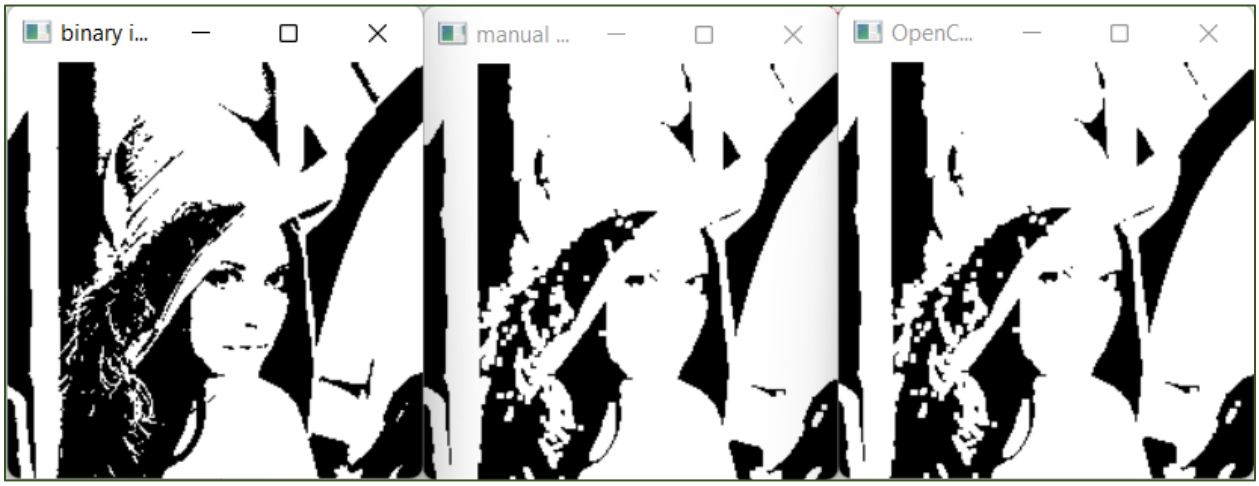
Toán tử Black-Hat: GSblackhat

(mở rộng) Toán tử textural Segmentation độ xám: GStese

4. Kết quả thực hiện

Các hình ảnh được sắp xếp theo thứ tự ảnh nhị phân (ảnh độ xám) – Ảnh thuật toán tự cài đặt – Ảnh tạo bằng hàm của OpenCV (nếu có)

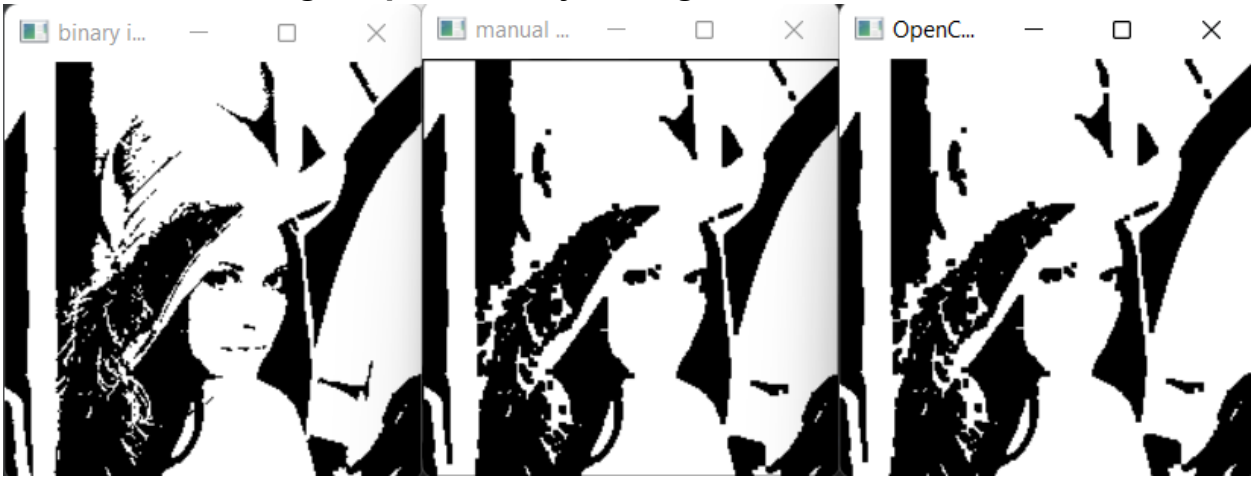
4.1. Toán tử giãn nở nhị phân (Binary Dilation)



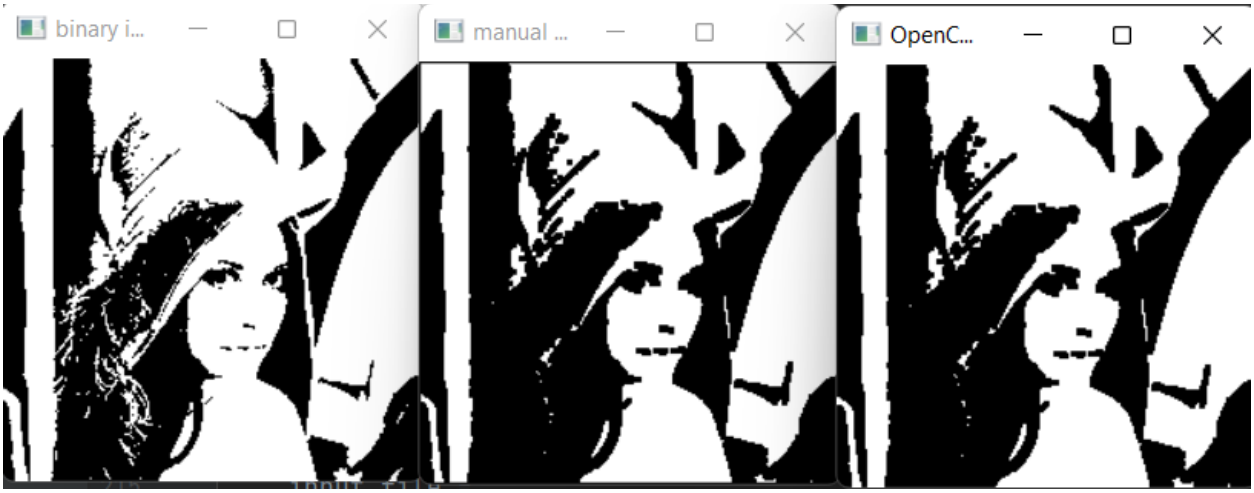
4.2. Toán tử co nhị phân (Binary Erosion)

4.3. Toán tử mở nhị phân (Binary Opening)

4.4. Toán tử đóng nhị phân (Binary Closing)



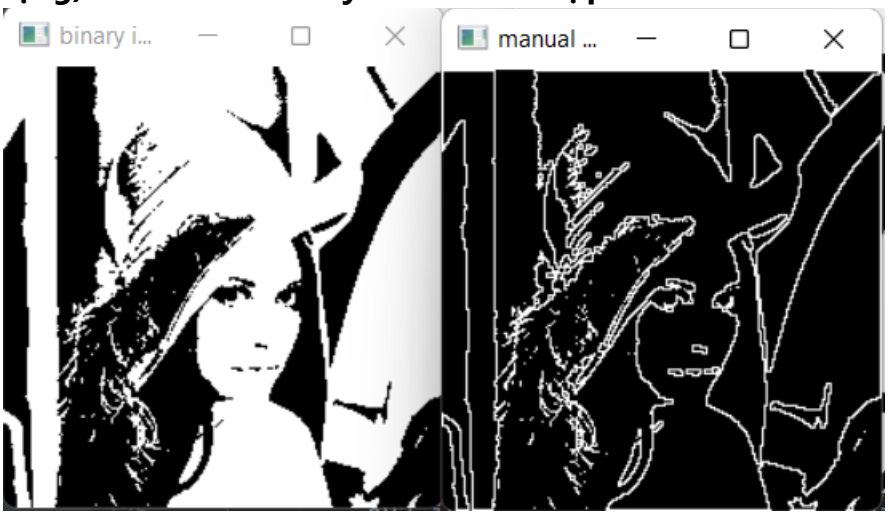
4.5. Toán tử Hit-or-Miss



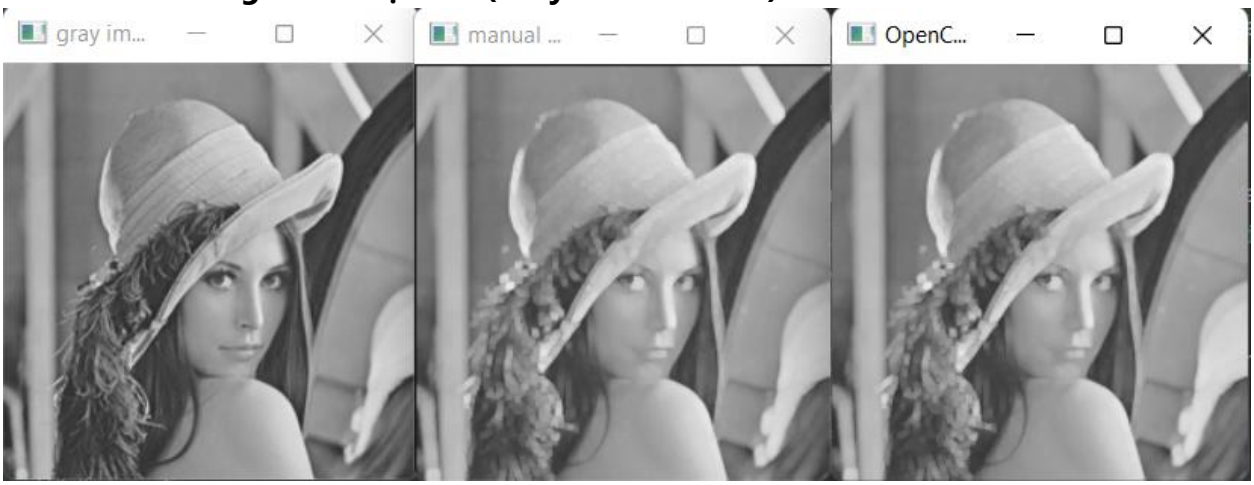
4.6. Thinning

(Thực hiện 48 vòng lặp hit or miss 8 mặt nạ)

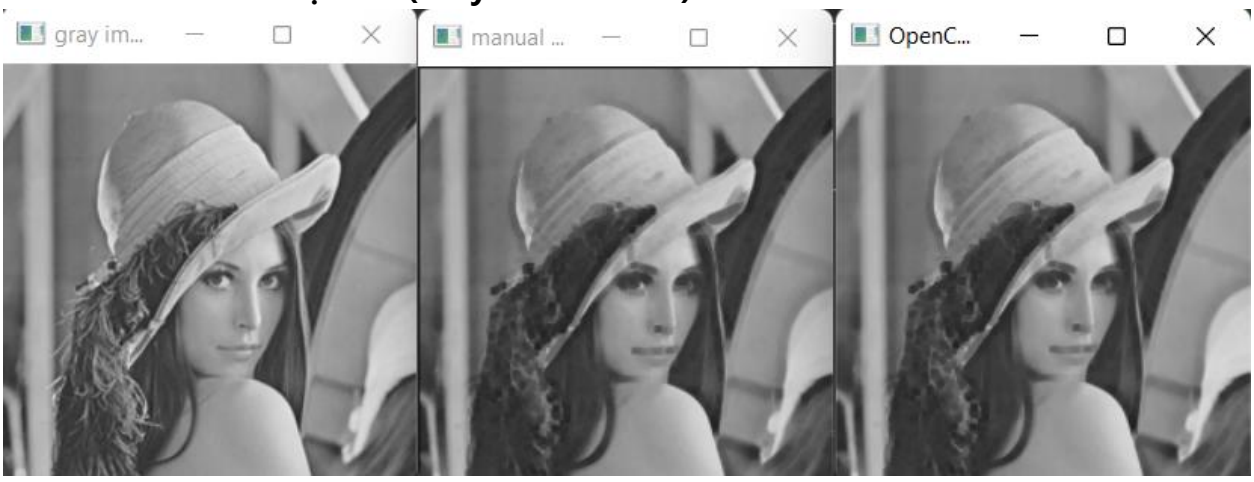
4.7. (mở rộng) toán tử Boundary Extraction nhị phân



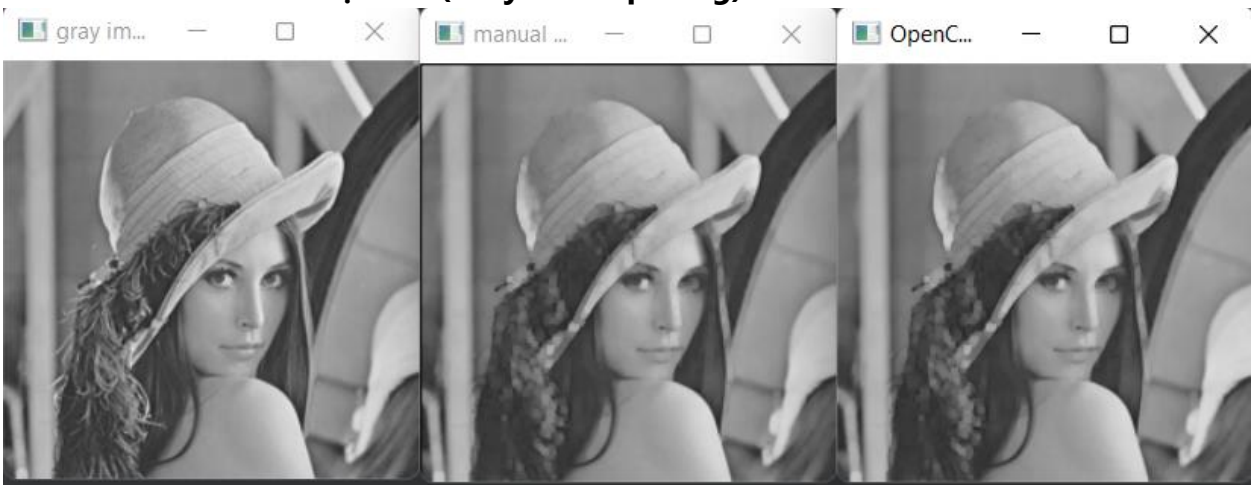
4.8. Toán tử giãn nở độ xám (Grayscale Dilation)



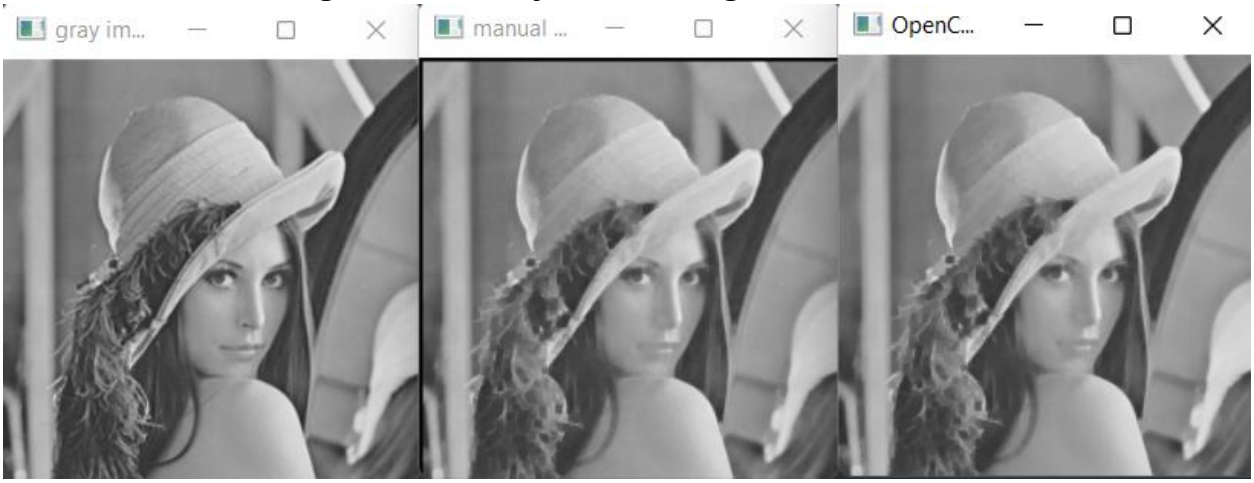
4.9. Toán tử co độ xám (Grayscale Erosion)



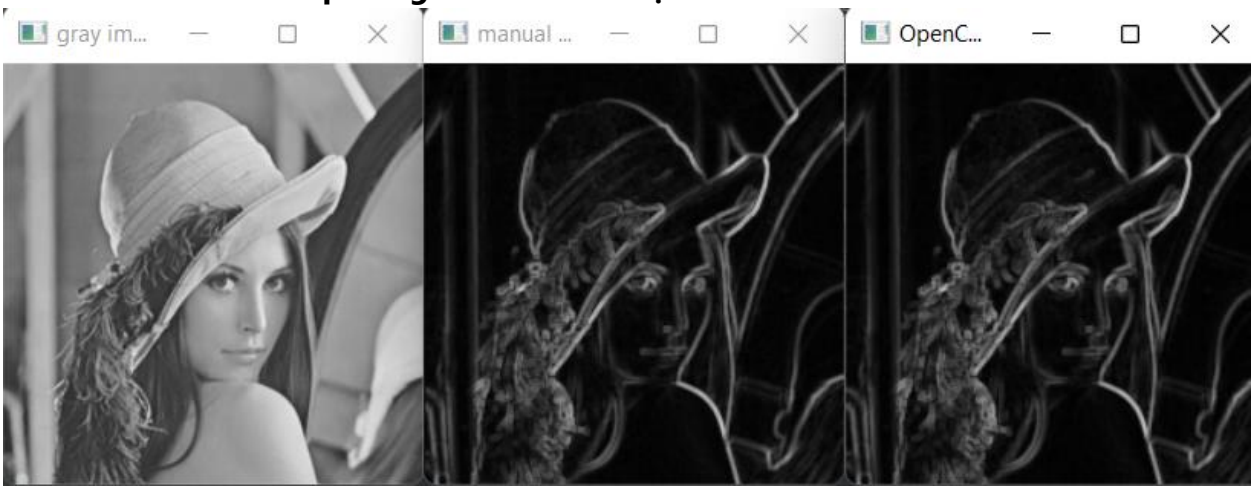
4.10. Toán tử mở độ xám (Grayscale Opening)



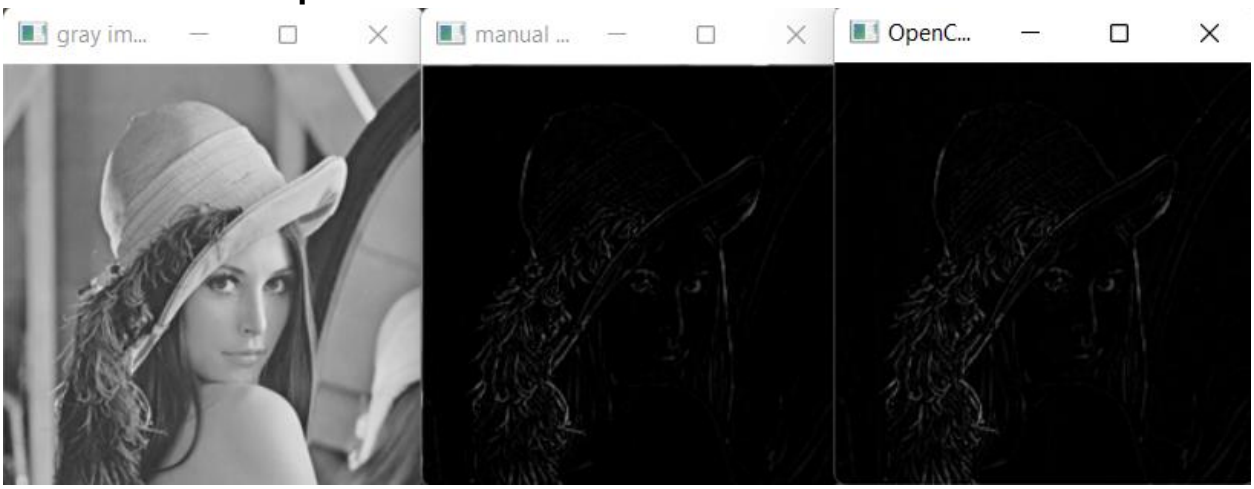
4.11. Toán tử đóng độ xám (Grayscale Closing)



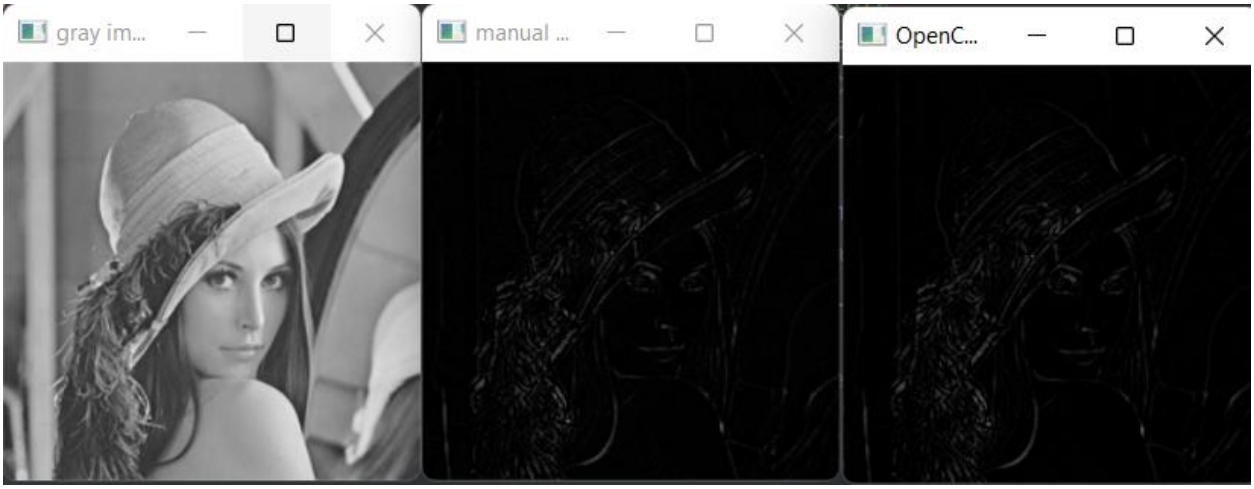
4.12. Toán tử Morphological Gradient độ xám



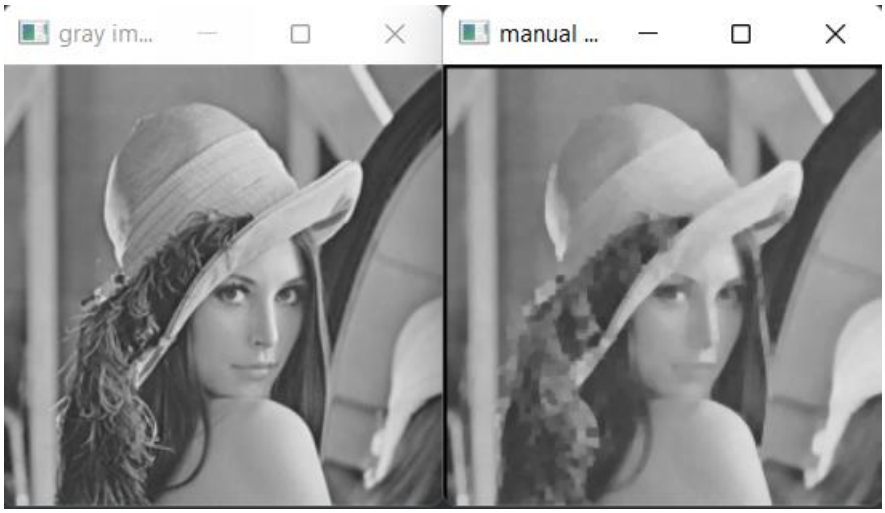
4.13. Toán tử Top-Hat



4.14. Toán tử Black-Hat



4.15. (mở rộng) Toán tử textural Segmentation độ xám



5. Nhận xét

Nhìn chung, các thuật toán tự cài đặt và hàm thư viện OpenCV đều cho ra kết quả trùng khớp với nhau, trừ trường hợp thực hiện toán tử thinning:

Về mặt thời gian thực hiện, các thuật toán của hàm thư viện nhanh hơn rất nhiều so với thuật toàn tự cài đặt, cá biệt nhất là trường hợp thinning, một ảnh kích thước lớn thinning bằng hàm thư viện chỉ mất vài giây, còn với thuật toán tự cài đặt, do có quá nhiều vòng for lồng vào nhau, do đó thuật toán tự cài đặt mất đến 40 phút để thực hiện cho ảnh lớn.

6. References

1. Digital Image Processing, third edition, Gonzalez

2. Slide bài giảng môn ứng dụng xử lý ảnh số và video số, PGS.TS Lý Quốc Ngọc

3. OpenCV, Hit-or-Miss:

https://docs.opencv.org/4.x/db/d06/tutorial_hitOrMiss.html

-HẾT-